



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,939	10/24/2003	James A. Lamb	200312021-1	2113

22879 7590 03/20/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER

ANYA, CHARLES E

ART UNIT	PAPER NUMBER
----------	--------------

2194

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

03/20/2008

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	Application No. 10/692,939	Applicant(s) LAMB ET AL.	
	Examiner Charles E. Anya	Art Unit 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3/ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12/14/08.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-11 and 14-32 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-11 and 14-32 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-11 and 14-32 are pending in this application.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. **Claims 1-3,5-9, and 17-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,484,309 B2 to Nowlin, Jr. et al. in view of U.S. Pat. No. 6,651,123 B1 to Hutchison et al.**

3. As to claim 1, Nowlin, Jr. teaches a computing device, comprising: an application layer (Windows 95/NT Application 20 Col. 2 Ln. 49 – 67); an operating system layer having a first type of operating system (Windows CE operating system/kernel 24 figure 2) and associated application program interfaces (APIs); and an interface module coupled between the application layer and the operating system layer (figure 2), wherein the interface module receives program instructions from a program in the application layer written for a second type of operating system (“...Win9X operating system...” Col. 2 Ln. 33 – 39, “...standard calling convention...” Col. 3 Ln. 4 – 18) and processes the instructions by directing the instructions through interpretation,

translation, and conversion to APIs that correctly execute the instructions (Translation Layer 22 Col. 2 Ln. 49 – 67, (“...ReQueryValueEx...” Col. 3 Ln. 35 – 40, “...translations...conversation...” Col. 3 Ln. 19 – 56); and wherein the interface module includes a discrete abstraction module having translation and conversion information therein (Translation Layer 22 Col. 2 Ln. 49 – 67).

Nowlin Jr. is silent with reference to the interface module receives program instructions from a program in the application layer written for a second type of operating system and processes the instruction through emulation and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

Hutchison teaches the interface module receives program instructions from a program in the application layer written for a second type of operating system and processes the instruction through emulation (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67) and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Nowlin, Jr. with the teaching of Hutchison because the teaching of Hutchison would improve the system of Nowlin, Jr. by using request emulator that are integrated within an application program ported to various different operating systems (Hutchison Col. 1 Ln. 51 – 54).

4. As to claim 2, Hutchison teaches the computing device of claim 1, wherein the interface module includes an operating system emulation module for emulating a number of operating system functions (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

5. As to claim 3, Hutchison teaches the computing device of claim 1, wherein the interface module emulates operating system functions and network server functions (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

6. As to claim 5, Hutchison teaches the computing device of claim 1, wherein the interface module emulates intelligent network server functions (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

7. As to claim 6, Hutchison teaches the computing device of claim 1, wherein the interface module has portions for emulating the operating system functions and the network server functions in discrete modules located within the interface module (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

8. As to claim 7, Nowlin, Jr. teaches the computing device of claim 1, wherein the interface module processes a program instruction by interpreting whether the instruction has to be processed further (“...ReQueryValueEx...” Col. 3 Ln. 35 – 40).

9. As to claim 8, Nowlin, Jr. teaches the computing device of claim 7, wherein the interface module converts a result received from the operating system layer such that the converted result is in a format that the application program can use to execute the instruction (Steps 44/50 Col. 3 Ln. 41 – 49).

10. As to claim 9, Nowlin, Jr teaches the computing device of claim 7, wherein the interface module translates the instruction received such that the operating system layer can execute the instruction (figure 3 Col. 3 Ln. 35 – 56, “...translate...” Col. 1 – 8).

11. As to claim 17, Nowlin, Jr. teaches a method of executing an application comprising: providing an application configured for an operating system; communicating instructions from the application to an interface module (figure 3 “...application calls...” Col. 3 Ln. 35 – 56) through interpretation, translation, and conversion (Translation Layer 22 Col. 2 Ln. 49 – 67, “...ReQueryValueEx...” Col. 3 Ln. 35 – 40, “...translations...conversation...” Col. 3 Ln. 19 – 56); and processing the instructions with the interface module to function with a different operating system (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 19 – 56) and wherein the interface module

includes a discrete abstraction module having translation and conversion information therein (Translation Layer 22 Col. 2 Ln. 49 – 67).

Nowlin, Jr. is silent with reference to communicating instructions from the application to an interface module through emulation and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

Hutchison teaches communicating instructions from the application to an interface module through emulation (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67) and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Nowlin, Jr. with the teaching of Hutchison because the teaching of Hutchison would improve the system of Nowlin, Jr. by using request emulator that are integrated within an application program ported to various different operating systems (Hutchison Col. 1 Ln. 51 – 54).

12. As to claim 18, Nowlin, Jr. teaches the method of claim 17, wherein processing the instructions from the application with the interface module includes using a list of instructions to be processed (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 19 – 56).

13. As to claims 19 and 21, Nowlin, Jr. teaches platform independence between multiple operating systems, however Nowlin, Jr., is silent with reference to the application being configured for a Linux based operating system/Unix operating system applications.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the platform independence of Nowlin, Jr. to include configuration/interoperability between Windows CE operating system and Linux operating system/Unix operating system applications to allow for increased compatibility.

14. As to claim 20, Nowlin, Jr. teaches the method of claim 17, wherein the application is configured for a Windows based operating system (“...Win9X...” Col. 2 Ln. 33 – 67).

15. As to claim 22, Nowlin, Jr. teaches the method of claim 17, wherein the method further includes identifying instructions to be translated by the interface module (“...translations...” Col. 3 Ln. 20 – 25, Ln. 35 – 56).

16. As to claim 23, Nowlin, Jr. teaches a method of executing an application (“...Win9X...” figure 2) configured for a platform having first type of operating system (“Windows operating system...” Col. 2 Ln. 33 – 67) on a platform having a second type

of operating system (Windows CE operating system/kernel 24 figure 2) comprising: communicating instructions from the application to an interface module, the application configured for a first type of operating system (figure 3 "...application calls..." Col. 3 Ln. 35 – 56); interpreting the instructions from the application with the interface module through interpretation, translation, and conversion ("...ReQueryValueEx..." Col. 3 Ln. 35 – 40, "...translations...conversation..." Col. 3 Ln. 19 – 56, "...translations..." Col. 3 Ln. 19 – 25, "...converts..." Col. 3 Ln. 35 – 40); communicating the instructions from the interface module to an operating system that is the second type of operating system (Step 44 Col. 3 Ln. 41 – 44) and wherein the interface module includes a discrete abstraction module having translation and conversion information therein (Translation Layer 22 Col. 2 Ln. 49 – 67).

Nowlin Jr. is silent with reference to interpreting the instructions from the application with the interface module through emulation and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

Hutchison teaches interpreting the instructions from the application with the interface module through emulation ("...file lock emulator..." Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67) and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein ("...file lock emulator..." Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Nowlin, Jr. with the teaching of Hutchison because the teaching of Hutchison would improve the system of Nowlin, Jr. by using request emulator that are integrated within an application program ported to various different operating systems (Hutchison Col. 1 Ln. 51 – 54).

17. As to claim 24, Hutchison teaches the method of claim 23, wherein communicating instructions from the application to an interface module includes communicating instructions to an operating system emulation module within the interface module (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

18. As to claim 25, Hutchison teaches the method of claim 24, wherein interpreting the instructions includes directing an instruction from the operating system emulation module to an application program interface (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

19. As to claim 26, Hutchison teaches the method of claim 23, wherein communicating instructions from the application to an interface module includes communicating instructions to a network server emulation module within the interface module (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

20. As to claim 27, Nowlin, Jr. teaches the method of claim 23, wherein interpreting the instructions includes translating an instruction configured for the first type of operating system to an instruction configured for the second type of operating system (Col. 1 Ln. 45 – 52, figure 3 Col. 3 Ln. 35 – 56).

21. As to claim 28, Nowlin, Jr. teaches the method of claim 23, wherein interpreting the instructions includes converting a result configured for the second type of operating system to a result configured for the first type of operating system (Step 48 Col. 3 Ln. 44 – 49).

22. Claims 4,10,11,14-16 and 29-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,484,309 B2 to Nowlin, Jr. et al. in view of U.S. Pat. No. 6,651,123 B1 to Hutchison et al. and further in view of U.S. Statutory Invention Registration No. H1,921 to Fletcher et al.

23. As to claim 4, Hutchison and Nowlin, Jr. are silent with reference to the computing device of claim 1, wherein the interface module emulates home location register functions.

Fletcher teaches the computing device of claim 1, wherein the interface module emulates home location register functions (“...interface...” Col. 11 Ln. 27 – 58).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Hutchison and Nowlin, Jr. with the teaching of Fletcher because the teaching of Fletcher would improve the system of Hutchison and Nowlin, Jr. by allowing for a central database that contains details of service subscribers that are authorized to use phone service core network.

24. As to claim 10, Nowlin, Jr. teaches a system architecture, comprising: an operating system layer having a first type of operating system; and an interface module (Translation Layer 22) to interface the application (Windows 95/NT Application 20) designed for a second type of operating system (Win9X operating system) with the first type of operating system (Windows CE operating system/Kernel 24 (figures 2/3) through interpretation, translation, and conversion (Translation Layer 22 Col. 2 Ln. 49 – 67, “...ReQueryValueEx...” Col. 3 Ln. 35 – 40, “...translations...conversation...” Col. 3 Ln. 19 – 56); and wherein the interface module includes a discrete abstraction module having translation and conversion information therein (Translation Layer 22 Col. 2 Ln. 49 – 67) and a connection for connecting the computing device to a publicly switched telephone network (PSTN) (“...wireless connection...” Col. 2 Ln. 21 – 26).

Nowlin, Jr. is silent with reference to a computing device including an application layer having a home location register application thereon and an interface module to interface the application designed for a second type of operating system with the first type of operating system through emulation and a discrete operating system emulation

module in communication with the discrete abstraction module and having interpretation information therein.

Hutchison teaches an interface module to interface the application designed for a second type of operating system with the first type of operating system through emulation (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67) and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Nowlin, Jr. with the teaching of Hutchison because the teaching of Hutchison would improve the system of Nowlin, Jr. by using request emulator that are integrated within an application program ported to various different operating systems (Hutchison Col. 1 Ln. 51 – 54).

Fletcher teaches a computing device including an application layer having a home location register application thereon (Col. 8 Ln. 37 – 38, Software Entities 312 Col. 11 Ln. 27 – 58).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Hutchison and Nowlin, Jr. with the teaching of Fletcher because the teaching of Fletcher would improve the system of Hutchison and Nowlin, Jr. by allowing for a central database that contains details of service subscribers that are authorized to use phone service core network.

25. As to claim 11, Nowlin, Jr. teaches the system architecture of claim 10, wherein the interface module has a number of modules to translate instructions between the operating system layer and the application layer (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 19 – 56).

26. As to claim 14, Hutchison teaches the system architecture of claim 13, wherein the operating system emulation module has translation and interpretation information therein (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

27. As to claim 15, Hutchison and Nowlin, Jr as modified by Fletcher teaches the system architecture of claim 10, wherein the system architecture further includes an operating system emulation module to direct an instruction from the home location register application to an application program interface (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 19 – 56).

28. As to claim 16, Nowlin, Jr teaches the system architecture of claim 10, wherein the system architecture further includes a number of component modules that can interface between an application designed for a second type of operating system and the operating system layer having a first type of operating system.

29. As to claim 29, Nowlin, Jr. teaches a computer readable medium having a set of computer executable instructions thereon for causing a device to perform a method, comprising: communicating instructions from a application to an interface module (Step 40 Col. 3 Ln. 35 – 56), the application configured for a first type of operating system (Win9X operating system); processing the instructions from the telecommunication application with the interface module through interpretation, translation, and conversion (Translation Layer 22 Col. 2 Ln. 49 – 67, “...ReQueryValueEx...” Col. 3 Ln. 35 – 40, “...translations...conversation...” Col. 3 Ln. 19 – 56); wherein the interface module includes a discrete abstraction module having translation and conversion information therein (Translation Layer 22 Col. 2 Ln. 49 – 67) and communicating the instructions from the interface module to an operating system that is a second type of operating system (Step 44 Col. 3 Ln. 41 – 45)

Nowlin, Jr. is silent with reference to communicating instructions from a telecommunications application to an interface module, the telecommunication application configured for a first type of operating system and processing the instructions from the application with the interface module through emulation and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information therein.

Hutchison teaches processing the instructions from the application with the interface module through emulation (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67) and a discrete operating system emulation module in communication with the discrete abstraction module and

having interpretation information therein (“...file lock emulator...” Col. 1 Ln. 51 – 67, figure 2 File Lock emulator 203 Col. 10 Ln. 17 – 38, Ln. 47 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Nowlin, Jr. with the teaching of Hutchison because the teaching of Hutchison would improve the system of Nowlin, Jr. by using request emulator that are integrated within an application program ported to various different operating systems (Hutchison Col. 1 Ln. 51 – 54).

Fletcher teaches communicating instructions from a telecommunications application to an interface module, the telecommunication application configured for a first type of operating system (Col. 8 Ln. 37 – 38, Software Entities 312 Col. 11 Ln. 27 – 58).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Hutchison and Nowlin, Jr. with the teaching of Fletcher because the teaching of Fletcher would improve the system of Hutchison and Nowlin, Jr. by allowing for a central database that contains details of service subscribers that are authorized to use phone service core network.

30. As to claim 30, Nowlin, Jr. teaches the computer readable medium of claim 29, wherein communicating instructions from an application to an interface module includes communicating to an abstraction module within the interface module (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 35 – 56).

31. As to claim 31, Nowlin, Jr. teaches the computer readable medium of claim 29, wherein communicating instructions from an application to an interface module includes communicating instructions to a component module within the interface module (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 35 – 56).

32. As to claim 32, Nowlin, Jr. teaches the computer readable medium of claim 29, wherein the method further includes identifying instructions to be converted by the interface module (Translation Layer 22 Col. 2 Ln. 49 – 51, Col. 3 Ln. 35 – 56).

Response to Arguments

Applicant's arguments filed 12/14/07 have been fully considered but they are not persuasive.

Applicant argues in substance that (1) Nowlin, Jr. prior art does not teach an interface module that includes a discrete abstraction module having translation and conversion information and a discrete operating system emulation module in communication with the discrete abstraction module and having interpretation information and (2) the Hutchison prior art teaches away from the invention because it is exclusively directed to incompatible file locking/unlocking that occurs when a application programming interface (API) is run on different operating systems.

The Examiner respectfully traverses Applicant's arguments:

As to point (1), the Nowlin, Jr. prior art discloses a system for allowing applications designed to execute on a particular operating system to run on another

operating system different the particular operating system. The translation layer (Translation Layer 22) sandwiched between the application layer and the operating system layer allows the applications to communicate with the different operating systems. The translation layer abstracts the communication between the application layer and the operating system layer by hiding the implementation details of a particular set of functionality in the operating system layer and allowing for translation/conversion of application request/call such that the operating system layer would understand the application request/call. The translation layer includes files/information (CoreDLL.DLL 24/Kernel32.DLL 27, User32.DLL 28, GDI32.DLL 29 and others 33) that provides for the application request/call translation/conversion.

As to point (2), the Hutchison prior art discloses a file locking emulator between an application program and an operating system. The file locking emulator comprises an application program interface and a file lock supervisor. The file locking emulator receives the file locking requests from the application program and generates file requests and file lock query commands. Only code in the file request emulator needs to be rewritten when an application program is ported to different operating systems sharing incompatible but executable processes and in this manner code of the application program does not have to be rewritten. The file locking emulator including the application program interface and the file lock supervisor allows application program written for an operating system platform to be ported to run on a different operating system hence the Hutchison prior art is analogous to both the Nowlin Jr. prior art and the instance invention as claimed.

As for Applicant's argument that the Hutchison prior art is exclusively directed to incompatible file locking/unlocking occurrences, the program instructions as claimed and received by the interface module does not exclude file locking requests/query and thus negates Applicant's argument. The type of request or instruction received at the interface module is irrelevant because firstly, the invention as claimed does not specify any particular type of request or instruction and secondly, the invention as claimed is directed to allowing application in an application layer to send request or instruction to the interface module and having the interface module translate/convert the request or instruction to the understanding of an operating system other than the operating system the application program is written for as both the Nowlin Jr. and Hutchison prior arts discloses.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles E. Anya whose telephone number is 571-272-3757. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

cea.